A Mobile Application to avoid Online Deception

COMP4801 – Interim Report

Lai Filbert Gibson (3035568013)
Lai Wai Yuet (3035568453)

Supervisor: Dr. Chim, Tat Wing

23rd January 2022
Abstract

In the 21st century, information technology is widely used around the globe. In light of the large amount of internet users in the world, cyber security is becoming more prominent for all internet users. The higher number of internet users leads to a higher success rate of online deceptions, which eventually leads to pecuniary losses or leakage of personal information. Therefore, in order to avoid online deception and raise public awareness of fraud or cyber pitfalls, this report introduces a project that aims to deliver a mobile application, ‘Online Guard,’ to avoid online deception by providing a safe browsing environment. Apart from adopting Google safe browsing service, the application will connect to large databases by leveraging the corresponding application programming interfaces in order to block reported malicious websites before browsing. Apart from reported websites, unreported malicious websites will also be detected by adopting advanced technologies like real-time deep analysis and artificial intelligence. To raise public awareness, news, information, and knowledge regarding cyber security will be introduced with a short quiz to examine the users. Unlike other similar mobile applications available in the market that require payment to unlock premium functions, Online Guard will be free to download with all functions provided. In the project, the browser page of the application has been implemented, which is able to block reported and suspicious phishing websites with an AI model developed in the development server. Among the three main functions that will be provided by Online Guard in the project proposal, two of them have been implemented, which met the target set for semester one. Apart from further improvements of the application and AI model, the remaining components to implement in the future are information and knowledge sharing pages as well as the quiz page.
Acknowledgement

We would like to express our great appreciation to the project supervisor, Dr. Chim, Tat Wing, for his valuable suggestions and guidance on this project. We would also like to extend our thanks to the Department of Computer Science, The University of Hong Kong, for providing guidance and the required resources for the project.
# Table of Contents

1 Introduction ..................................................................................8
  1.1 Background ..........................................................................8
  1.2 Objectives ............................................................................8
  1.3 Motivations ...........................................................................8
  1.4 Scope ....................................................................................9
  1.5 Outline of Report .................................................................10

2 Methodology ................................................................................11
  2.1 System Design .......................................................................11
  2.2 Blocking reported phishing websites .....................................12
    2.2.1 User-defined blocklist ....................................................13
    2.2.2 Google Safe Browsing Service ......................................16
    2.2.3 APIs for accessing large databases .................................16
  2.3 Scanning for suspicious websites ..........................................18
    2.3.1 APIs for analysing URL ................................................19
    2.3.2 AI model ......................................................................19
  2.4 Machine Learning Model .......................................................20
    2.4.1 Finding .........................................................................20
    2.4.2 Dataset .........................................................................20
    2.4.3 Model Training .............................................................21
    2.4.4 Performance .................................................................22
    2.4.5 Features Selection .........................................................22
  2.5 Information Sharing regarding cyber security .........................25
    2.5.1 Latest Cyber Security News ..........................................26
    2.5.2 Cyber Security practices to avoid online deceptions ..........27

3 Project Plan ..................................................................................28
  3.1 Current Status .......................................................................28
  3.2 Scope of Future Works .........................................................28

4 Project Limitations ........................................................................30

5 Conclusion ...................................................................................32

References ......................................................................................33
List of Figures

Figure 1.1: Pricing of the BlockSite Solutions ................................................................. 9
Figure 2.1: Overall System Design .................................................................................. 11
Figure 2.2: a part of the screenshot in the setting page .................................................... 13
Figure 2.3: Screenshot of the blocklist page ..................................................................... 13
Figure 2.4: Screenshot of browser page when website blocked by blocklist .................... 14
Figure 2.5: removing an item in blocklist ....................................................................... 15
Figure 2.6: Phishing website detected by Google Safe Browsing Service ......................... 16
Figure 2.7: A real phishing website detected by ‘URL Reputation API’ from URLVoid ...... 17
Figure 2.8: A real phishing website detected by ‘Malicious URL Scanner API’ from IP Quality Score ......................................................................................................................... 18
Figure 2.9: System mechanism to fulfil main function 1 and 2 ........................................ 19
Figure 2.10: Dividing a URL into different substrings ..................................................... 20
Figure 2.11: XGBoost performance on real-world data .................................................. 22
Figure 2.12: Feature importance using MDI on all features ............................................ 23
Figure 2.13: XGBoost performance on real-world data on all features ............................. 24
Figure 2.14: Feature importance using MDI on selected features .................................... 25
Figure 2.15: System mechanism to fulfil main function 3 ................................................. 26
Figure 2.16: Prototype of News page .............................................................................. 27
Figure 4.1: Feature importance using MDI on UCI dataset ............................................. 30
List of Tables

Table 2.1: Performance on the four learning models.............................................. 20
Table 3.1: Planning on future work................................................................. 29
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>APP</td>
<td>Application</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>HKD</td>
<td>Hong Kong Dollar</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Hypertext Transfer Protocol Secure</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>IPQS</td>
<td>IP Quality Score</td>
</tr>
<tr>
<td>MDI</td>
<td>Mean Decrease in Impurity</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>UX</td>
<td>User Experience</td>
</tr>
<tr>
<td>WEB</td>
<td>Website</td>
</tr>
</tbody>
</table>
1 Introduction
This section will cover the background, objectives, motivation and scope of the project. An outline of the report will also be introduced.

1.1 Background
In the 21st century, information technology is widely used around the globe, especially the application of the World Wide Web for website browsing. According to the research conducted by Statista [1], in January 2021, there were around 4.66 billion active internet users in the world. However, online frauds are becoming more and more common in Hong Kong. There were more than 10,500 internet deception cases reported to the police force in 2020, and the technology-related crime had costed more than 2,900 million HKD loss [2]. The number of internet deception cases has an increasing trend. The number of online frauds reported in 2019 was only about 5,000 cases [2], which was doubled in 2020. It is an emergency issue to provide the community a secure and reliable cyber environment.

1.2 Objectives
Two main objectives are included in the project. The first objective is to enhance the cyber security sense and raise public awareness of online deception. This objective will be achieved by sharing knowledge and information about cyber pitfalls. The second objective is to provide a safer internet environment for the public. This will be achieved by filtering the websites, so that the access towards phishing or suspicious websites will be blocked so the public will be protected from illegal websites.

1.3 Motivations
The motivation of this project is to provide a safer internet environment for users and increase the public awareness of information security. In the mobile environment, there are similar applications, such as ‘BlockSite’ on Google Play Store [3]. However, like many other similar applications, functions are limited and users are required to pay subscription fees in order to unlock the complete protection from the application [4]. Figure 1.1 shows the multiple features available only in the ‘Unlimited Plan’ which requires subscription fee, like
an unlimited blocking list, custom block pages and password protections. Hence, our team aims to provide a mobile application to avoid online deception but without charging the users.

![Diagram](image.png)

Figure 1.1: Pricing of the BlockSite Solutions [4]

### 1.4 Scope

Our team introduces Online Guard, which is an Android mobile application providing cyber protection to users and increasing their awareness. The app is developed using Android Studio as IDE, which will be free to download with all functions provided freely. Online Guard tries to achieve the functions by three main methods. First, the app should be able to block the access towards reported phishing websites when the users try to access them. Second, the app should be able to detect potential phishing websites using different technologies, including APIs and machine learning models. Users should be warned if they try to access suspicious websites. Lastly, the app should be able to provide news and information regarding cyber security so that the users can be updated to avoid online deception.
1.5 Outline of Report
The remaining of the paper will start with discussing the methodology of the project in Section 2. In section 2.1, the paper will introduce the system design of the project, followed by the function to block reported phishing websites in section 2.2. The paper will then cover the function to scan suspicious websites. Next, the paper will discuss the machine learning model used, including the related experiment and the findings, followed by introducing the last main function – information sharing regarding cyber security. In section 3, the current progress and the future planning will be discussed. The paper closes by describing the limitation faced in section 4 and concludes in section 5.
2 Methodology
This section introduces the overall system design and the three main functions provided by the mobile application. For each main function, the working mechanism, methodologies and justifications will be discussed, followed by the discussion regarding the machine learning model, such as finding and performance.

2.1 System Design
Figure 2.1 shows the overall system design which is constructed by the mobile application, development server and malicious URL detectors.

![Overall System Design Diagram]

The mobile app contains different pages, including (1) Browser page to provide a browser with safe browsing experience, (2) Blocklist page which allows users to view and edit a locally stored blocklist which contains the domain name and URL that users want to block in the browser, (3) News page which provides the latest cyber security news, (4) Info page which provides useful cyber security practices or tips to avoid online deceptions, (5) Quiz page which examining users about the knowledge in info page.
The development server contains (1) an HTTP Server to communicate with the mobile app by handling HTTP requests, (2) machine learning model to detect potential phishing websites, (3) Website crawling program to fetch the latest news from the internet and store them into a local database, (4) MongoDB database that stores the title, URL and source of articles regarding cyber security.

The mobile application will provide three main functions, including (1) Blocking reported phishing websites, (2) Scanning for suspicious websites, (3) Information Sharing regarding cyber security. For each main function, the working mechanism, methodologies and justifications will be discussed in the following sections.

2.2 Blocking reported phishing websites

Blocking reported phishing websites is one of the main functions provided by the mobile application. Before users browse a website using the browser page in Online Guard, the URL will be checked using multiple technologies, including (1) User-defined blocklist, (2) Google safe browsing service, (3) APIs from two platforms, including URLVoid and IP Quality Score.

If the website is detected as a phishing website, the website access will be blocked with a pop-up window generated for showing the details, like the screenshot shown in figure 2.4. Users can choose to cancel the browsing by pressing the ‘Cancel’ button on the pop-up window. Alternatively, users can choose to browse the website by pressing the ‘Still Go’ button. Furthermore, users can choose between blocking the website directly or showing a pop-up window in the setting page. Figure 2.2 shows the related part in the setting page that users can decide the behaviour of the app when a particular technology detects a website as a phishing website.
2.2.1 User-defined blocklist
Users can view and edit a user-defined blocklist in the blocklist page. The blocklist contains domain names or URLs which will be blocked in the browser page. For example, figure 2.2 shows a screenshot of the blocklist page that a domain name ‘youtube’ and an URL ‘https://www.cuhk.edu.hk/English/index.html’ exist in the blocklist.
If the website that the user is going to browse in the browser exists in the blocklist, it will be blocked with a pop-up window showing the reason of blocking, like the screenshot shown in figure 2.4.

Figure 2.4: Screenshot of browser page when users go to a website with domain name exists in the blocklist

For adding domain name or URL to the blocklist, users can type in the text field and press the add button. For deleting item, users can tap the item and press the confirm button on the pop-up window, like figure 2.5. The application will update the change to the user interface and the local device immediately.
The blocklist serves as a security patch. If there are unreported phishing websites that Online Guard cannot detect, users can still report the websites as phishing. Initially, our team planned to locate the blocklist in the development server. Users can report phishing websites to the server so that the websites will be blocked globally such that other users cannot access the website also. However, there will be a potential problem that some users may mistakenly or maliciously add safe websites to the blocklist so other users will be affected too. For example, if a user mistakenly added a domain name of a legitimate website into the blocklist, other users will not be able to access the web page also. It can be solved by adding an investigation mechanism that every feedback provided by users will be investigated before adding it to the shared blocklist. Due to the limitation of resources, our team have finally decided to place the blocklist locally on the mobile device.
2.2.2 Google Safe Browsing Service
Google safe browsing service is also used to provide safer environment, which is a Google service that allows applications to check URLs against a constantly updated list maintained by Google [5]. The list contains reported unsafe web resources like phishing and deceptive sites.

There is a WebView component in the browser page in Online Guard. The WebView is configured to use Google safe browsing service for internet surfing. If the website is detected as a phishing website by Google safe browsing service, it will be blocked with a red background, like the screenshot shown in figure 2.6.

![Figure 2.6: Phishing website detected by Google Safe Browsing Service](image)

2.2.3 APIs for accessing large databases
Apart from user-defined blocklist and Google safe browsing service, two APIs are also leveraged to block reported phishing websites, including the APIs from two platforms - ‘URLVoid’ and ‘IP Quality Score’.
Firstly, Online Guard uses the ‘URL Reputation API’ from the platform ‘URLVoid’ to check the reputation of URL from more than 25 engines, such as OpenPhish, PhishTank and Phishstats [6]. Also, the API can detect for unreported phishing websites which will be further discussed in section 2.3.1. If the website is detected as a phishing website by the API, it will be blocked with a pop-up window showing (1) the reason for blocking, (2) the risk score calculated by URLVoid, (3) the number of engines that returned the website as a phishing website, (4) website title which helps users to decide whether to proceed or not. Figure 2.7 shows a screenshot of a real phishing website blocked by the API.

![Figure 2.7](image.png)

Figure 2.7: A real phishing website detected by ‘URL Reputation API’ from URLVoid

Secondly, Online Guard uses the ‘Malicious URL Scanner API’ from the platform ‘IP Quality Score’ to identify phishing, malware URL and parked domains [7]. Similar to the API from URLVoid, this API uses a live threat intelligence feed to detect zero-day phishing URLs, which will be further discussed in section 2.3.1. If the website is detected as a phishing website by the API, it will be blocked with a pop-up window showing (1) the reason of blocking, (2) risk score calculated by IP Quality Score, (3) reasons of blocking provided...
by the API, (4) website category which helps users to decide whether to proceed or not. Figure 2.8 shows a screenshot of a real phishing website blocked by the API.

Figure 2.8: A real phishing website detected by ‘Malicious URL Scanner API’ from IP Quality Score

2.3 Scanning for suspicious websites
Apart from reported phishing websites, unreported phishing websites should also be detected to enhance the security level. Apart from the two APIs aforementioned, Online Guard uses the machine learning model in the development server to scan for suspicious websites.

Figure 2.9 shows the system mechanism in providing a safe internet environment. When the users browse a website in the browser page, the URL will be checked against (1) a user-defined blocklist stored locally, (2) Malicious URL detectors, including Google safe browsing service, URLOvoid API and IPQS API, (3) the machine learning model in the development server that Online Guard gets the analysis of the machine learning model from the HTTP server by GET request.
2.3.1 APIs for analysing URL
Two APIs mentioned in section 2.3.3 can be used for analysing zero-day phishing links.

For ‘URL Reputation API’ from the platform ‘URLVoid’, it uses thousands of smart internal rules to detect potential malicious URLs [6]. Deep analysis on the URL is performed, such as URL content, pattern, domain name and HTTP headers.

For ‘Malicious URL Scanner API’ from the platform ‘IP Quality Score’, it uses a live threat intelligence to detect zero-day phishing links in real time [7]. Risk score will be provided after real-time risk analysis. Also, the intelligence keeps improving by feeding threat data from their clients.

2.3.2 AI model
Apart from APIs, our teams have also developed an AI model in the development server to detect for zero-day phishing links. When users browse a website, Online Guard sends the URL to the HTTP server by GET request and gets the prediction result of the machine
learning model in the JSON message returned by the server. Details of machine learning and machine learning model will be discussed deeply in section 2.4.

2.4 Machine Learning Model
In the project, machine learning model is trained and used to perform potential phishing detection. When Users access an URL, the URL will be passed to the model and perform prediction. Features are extracted from the URL and the website, then be inputted into the model to perform prediction. Warning will be given if the website is predicted as a phishing website. Serval models were trained and the model with the best performance was chosen as the model to be implemented.

2.4.1 Finding
Our team had performed experiments to pick the best model. The training progress and the decision made will be introduced in the following sections. The following sections will first introduce the dataset used, followed by discussing the training of the model. This section will then evaluate the performance of the model and cover the feature selection.

2.4.2 Dataset
The dataset used to train the model comes from the paper ‘Datasets for phishing websites detection’ [8]. The dataset used contains features extracted from a total of 30,647 confirmed phishing URLs that were collected from PhishTank. It also contains features extracted from a total of 58,000 legitimate website URLs that were collected from Alexa. The unbalanced variant here is to simulate the real-world situation [8], where the number of legitimate websites is much larger than the number of phishing websites.

Figure 2.10: Dividing a URL into different substrings [8]

The features in the dataset can be divided into six main groups. Figure 2.10 shows how a URL is divided into substring groups in the dataset. A URL is divided into four parts
according to their roles and functions. The first five groups are the attributes based on the whole URL string and the four substrings. These five groups of features are string properties, such as the number of specific symbols presented, such as `@`, `/`, and the number of arguments presented. The last group of features are the URL resolved and external service features of the website. Features such as the number of redirections, the domain look-up time and the Time-To-Live of the hostname are included in this group.

### 2.4.3 Model Training

In the model training stage, three models were trained to fight the baseline performance, which was the performance of logistic regression. The three models built are support vector machine (SVM), extreme gradient boosting (XGBoost) and random forest. Before the learning models were fed with data, our team first processed the data. There are columns that all the values under the column are the same. Those columns were first removed from the dataset as those columns cannot provide information in helping the classification. The columns containing continuous values also underwent feature scaling before the data were fed into SVM and logistic regression model. The data were normalized into the same range in order to have a faster iteration and a better convergence [9]. Feature selection was also performed to improve the performance, which will be discussed later in section 2.4.4. The dataset was separated into training set and testing set by a ratio of 80% to 20%. The trained model was evaluated on the testing set.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>0.6753</td>
<td>0.781</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.8931</td>
<td>0.905</td>
</tr>
<tr>
<td>XGBoost</td>
<td>0.8776</td>
<td>0.888</td>
</tr>
<tr>
<td>SVM</td>
<td>0.7732</td>
<td>0.807</td>
</tr>
</tbody>
</table>

Table 2.1 Performance on the four learning models

The performances of the four models were accessed by their overall accuracy and recall rate on the phishing class. The recall value shows the percentage of the phishing websites that the model can correctly classify as phishing. Our team focus more on recall value instead of the precision value because the consequence of the model to miss a phishing website is more
serious than incorrectly classifying a legitimate website as phishing. Table 2.1 shows the overall accuracy and the recall value of the four models on the selected features. The baseline performance, logistic regression, had an accuracy of 67.53% and a recall rate of 78.1%. The random forest and XGBoost had the best performance among the four models, both having an accuracy of more than 87% and a recall rate of more than 88.8%. Their performances are very similar because their architectures are very similar [10].

2.4.4 Performance
The performance of the model was tested with the real-world data. Our team collected 198 phishing websites from Phishtank [11]. 170 legitimate websites were also collected from Alexa and our team. The features of these websites were extracted and inputted to the model that has the best result, the XGBoost model, to perform prediction. The performance of the model was evaluated.

![Figure 2.11: XGBoost performance on real-world data](image)

Figure 2.11 shows the three matrices regarding the performance of the XGBoost model, which are the confusion matrix, precision matrix and recall matrix. From the recall matrix, the true positive rate was 73.2%, meaning the model can classify 73% of the phishing websites as phishing. Meanwhile, the false positive rate was 30.6%. It means the model will incorrectly classify 30% of the legitimate website as phishing website. The model had an overall accuracy of 71.46%.

2.4.5 Features Selection
The importance of each feature in the dataset was evaluated to help us have a better understanding. The importance of feature was calculated by the mean decrease in impurity (MDI). The MDI of each feature inside the random forest were measured. Furthermore, the
deviation of MDI of the features were also measured. Figure 2.12 shows the most important features among all the features measured by MDI. The blue bar shows the MDI value of the features among all the trees inside the forest, and the black line indicates the deviation of decrease in impurity of the feature among all trees. The result shows the length of directory string had the highest MDI, followed by the domain activation time and the length of the URL. The figure shows that the performance of the features was not very satisfying, only the two best features had MDI over 0.1, while other features’ MDI are 0.05 or even lower. The result also shows the deviations of the substring attributes are higher. Most of the deviation values of substring attributes had deviation covering negative values. This meant the substring attribute cannot promise a decrease in impurity.

The performance of the model without feature selection was not satisfying. The models are trained on all the features. The model had a well performance on the dataset. The two best models, XGBoost and random forest, both had overall accuracy and recall rate of more than 95%. However, the performances of the models dropped rapidly on the real-world dataset.
Figure 2.13: XGBoost performance on real-world data on all features

Figure 2.13 shows the three matrices regarding the performance of the XGBoost model using all the features in the dataset. From the recall matrix, the true positive rate was 70.2%. However, the false positive rate was 40.6%. The performance is not satisfying due to the high false positive rate. The model performed well in the dataset but did not perform well in the real-world data. One of the reasons may be the dataset does not reflect the real-world situation. From the feature importance figure, most of the important features were the substring attributes. However, these features may be just noisy, as most of them had high deviation.

To improve the performance, most of the substring attributes were removed, while all the URL resolving features and the external service features were kept. The feature importance was evaluated on the newly selected features.
Figure 2.14: Feature importance using MDI on selected features

Figure 2.14 shows the feature importance of the selected features. The performance in decreasing the impurity had been increased. The best three features had MDI of more than 0.1, even 0.2. There was also number of features having MDI of more than 0.5. The deviations of each feature also decreased significantly. Almost none of the features cover negative values. This implies the dataset had fewer noisy feature columns after feature selection. The model result mentioned in section 2.4.3 was trained by the selected features, where improvements were made in comparison to the performance using all features.

2.5 Information Sharing regarding cyber security

The last main function to be provided by Online Guard is information sharing regarding cyber security, which can be divided into two parts – (1) sharing the latest cyber security news from multiple sources, (2) sharing common cyber security practices to avoid online deceptions.
2.5.1 Latest Cyber Security News
Online Guard provides latest cyber security news in the news page. Figure 2.15 shows the related system mechanism when users browse the news. When users load the news page, the application will fetch the title, source and URL of the news article from the database in the development server through HTTP server. Afterwards, the app will display the title and source of news in a list, like the prototype shown in figure 2.16. When users click an item, the corresponding URL will be loaded for the users to read the news article.

In order to collect and provide the latest news, there will be a website crawling program in the development server running regularly to fetch the title, source and URL of news from the internet and store the data into the database.

Figure 2.15: System mechanism to fulfil main function 3
The website crawling program is placed in the server. The mobile app fetches the required data from the database directly when the news page is loaded. It reduces the loading time as the time of reading data from the database is shorter than crawling and processing data from multiple websites.

2.5.2 Cyber Security practices to avoid online deceptions
Apart from latest cyber security news, some common techniques or practices to avoid online deceptions can be shared also, such as ‘User password should have a minimum length of 8 characters.’

The info page will contain the cyber security practices or tips to avoid online deceptions. Meanwhile, users can attend a short multiple choices quiz in the quiz page for knowledge consolidation. After finishing the quiz, the application will mark the answer of users and show all the correct answers. A score will be calculated to act as a reference so that users will know how well they know about the cyber security practices.
3 Project Plan
This section will discuss the progress and the future planning of the project.

3.1 Current Status
The overall progress had covered about 55% of the whole project. The target set for semester one was met.

First, the server of the mobile application and the database were set up. The server is able to handle requests from the app and query the database, while the tables were created in the database so that it is able to store phishing website data and news. Also, AI model was developed in the server such that it can perform prediction of the URL sent by the mobile apps to the HTTP server by GET request. Second, the mobile app was developed with browser page, blocklist page, setting page and help page finished with consideration of UX design, such as responsive and self-descriptive.

Among the three main functions that planned to be provided by the mobile app, two of them were implemented – (1) blocking reported phishing websites, (2) scanning for suspicious websites. The browser in the mobile app is able to block reported and unreported phishing websites using blocklist, APIs and AI.

3.2 Scope of Future Works
The work remaining includes improving the performance of the machine learning model and completing the remaining main functions, including the implementation of the news page, information page and the quiz page. The application will also be further improved.
Table 3.1 shows the planning of the deliverables to be implemented in the next phase. In January, improvements will be made on the mobile app and the machine learning model. Our team targets an over 75% of accuracy and recall rate. The user experience of the app will be the focus when improving the app. The main function 3, information sharing regarding security, will be implemented in February and March. The news page and the information page of the mobile app will be completed in February, while the quiz page will be finished in March. The corresponding functions in the server will be updated at the same time. App testing will also be performed in March to check whether all the functions in the app can be executed correctly without bugs. In April, all the deliverables will be implemented, and the project will be finalized. The final report will also be completed in April.
4 Project Limitations
The most significant limitation our team faced was the lack of phishing dataset resources. There are not many available choices for dataset. However, some datasets cannot reflect the real-world situation. Apart from the dataset chosen to train the model mentioned in section 2.4.1, our team performed experiments on another dataset ‘Phishing Websites Data Set’ from the UCI repository [12]. This dataset extracts different features from websites. Figure 4.1 shows the mean decrease in impurity (MDI) in random forest of the features in the dataset. The feature with the best performance was the SSL_final_State, which is whether the website has a valid HTTPS certification or not. This means the HTTPS certification have a relative heavy on classifying phishing websites.

![Feature importance using MDI on UCI dataset](image)

A valid certification in this dataset is defined as a certification issued by a trusted issuer and the time to expire is longer than one year. The trained model in this dataset had performance of both accuracy and recall rate higher than 96%. However, in the dataset, 91.44% of the phishing URLs do not have any valid cert and 0.34% of the phishing URLs have cert but not valid. While in the crawled real-world data mentioned in section 2.4.3, only 37.9% of the URLs do not have any valid cert and all the remaining URLs, more than 60%, have cert that
is not valid. The difference leads to the high false positive rate and unsatisfying performance. To handle this problem, our team will try to create our own training dataset by crawling real-world phishing websites and legitimate websites, so that the dataset will be able to reflect the real situation.
5 Conclusion
This paper has presented the background, the method used, the current findings and the progress made in the project. This project is motivated by the arising cyber security risk and our team aimed to provide a solution to provide a safer internet environment. This paper has introduced a project to deliver a free mobile application and all functions will be delivered freely.

The application is able to block the access towards phishing websites or potential fraud websites. It uses APIs and blocklist to block the access towards known phishing websites while it uses machine learning model and APIs to detect unknown potential threats. By blocking the website access, online deceptions can be avoided in order to prevent the leakage of personal information or pecuniary losses. The application should also be able to share information regarding cyber security by crawling information online. There are also quizzes to further enhance the knowledge of the public.

Lastly, this paper has discussed the progress made by our team. The functions on blocking phishing websites and detecting potential fraud websites have been implemented already. The pages in the application corresponding to these functions are already built, with all the main configurations of the app completed. Moreover, the related functions have been implemented inside the server. The database has also been implemented. These implemented deliverables will be further improved continuously in the coming phases. By the end of the year, the last function, which is information sharing regarding cyber security, will be implemented. The related pages in the app and the functions in the server will be completed.
References


